Kotitehtävä 3

Öljy koostuu suurelta osalta hiilivetymolekyyleistä. Hiilivetymolekyylit koostuvat hiili- ja vetyatomeista. Kuhunkin hiiliatomiin liittyy 4 ja kuhunkin vetyatomiin 1 sidos. Kahden hiiliatomin välillä voi olla myös kaksois- tai kolmoissidos, jolloin atomien muiden sidosten määrä on vastaavasti pienempi. Öljyn hiilivetyjen pituus vaihtelee suuresti; esimerkiksi bensiini sisältää lähinnä hiilivetyjä, joissa on 5–10 hiiliatomia[1].

Hiilivetymolekyylin hiiliatomien muodostama rakenne voidaan esittää yhtenäisenä multiverkkona, jonka solmut ovat hiiliatomeja ja jonka kaaret vastaavat hiiliatomien välisiä sidoksia. Multiverkko on verkko, jossa saman kaaren toistuminen sallitaan. Hiilivetyjä mallinnettaessa minkään solmun asteluku ei saa ylittää neljää, ja kaksois- ja kolmoissidokset voidaan mallintaa laittamalla kahden solmun välille enintään kolme kaarta. Tämä malli ei selvästikään ota huomioon esimerkiksi kolmiulotteisen geometrian rajoituksia.

Olkoon $d(v)$ solmun v asteluku. Kun $G = (V, E)$ kuvaa ylläesitettyyn tapaan hiilivetymolekyyliä, jossa on $|V|$ hiiliatomia, molekyylissä on $\sum_{v \in V} (4 - d(v)) = 4|V| - 2|E|$ vetyatomia. Kaksi hiilivetyä ovat isomeerejä, jos sekä hiili- että vetyatomien lukumäärä on sama.

Tehtävä

Pidetään kahta hiilivetyä isomorfisina, jos niitä vastaavat multiverkot ovat isomorfiset. Laadi tietokoneohjelma ja generoi sillä mahdollisimman suurelle n ei-isomorfiset hiilivedyt, joissa on enintään n hiiliatomia. Ryhmittele ei-isomorfiset hiilivedyt isomerialuokkiin ja ilmoita, montako hiilivetyä kuhunkin isomerialuokkaan kuuluu. Ilmoita lisäksi kustakin isomerialuokasta, kuinka moni luokan hiilivety on syklitön. Palautuksia arvioitaessa arvioidaan pikemminkin menetelmien tehokkuutta kuin saavutettuja tuloksia; yletöntä laskentaresurssien käyttöä ei pidetä meriittinä. Noin 1 h CPU-aikaa on riittävä suuruusluokka.

Vihjeitä ja apuohjelma

Ongelman ratkaisussa tarvittaneen keino laskea multiverkon kanoninen muoto. Kurssin kotisivulla tiedostossa `kotit3.c` on C-kielinen aliohjelma multiverkon kanonisen muodon laskemiseksi:

```
void canon(int vertices, char *s, char *canon);
```

[1] http://www.lloydminsterheavyoil.com/refining.htm

`canon` laskee multiverkon kanonisen muodon. Syötemultiverkko määritellään antamalla parametrissä `vertices` sen solmujen määrä ja parametrissä `s` merkkijono, joka määrittää itse multiverkon. Solmut numeroidaan `1...vertices` ja merkkijono

```
s=b_2,1 b_3,1 b_3,2 b_4,1 b_4,2 b_4,3 b_5,1 ... b_vertices,vertices-1
```

missä `b_j,i` on merkki, joka kuvaa, montako kaarta solmujen i ja j välillä on. Merkki '0' tarkoittaa 0 kaarta, merkki '1' yhtä kaarta jne. Aliohjelma muuntaa syötemerkkijonon verkoksi, laskee sen kanonisen muodon ja kirjoittaa merkkijonoon `canon` kanonista muotoa vastaavan merkkijonon. Käyttäjän tulee huolehtia muistin varaamisesta `canon`-merkkijonolle.

Tämä aliohjelma puolestaan käyttää Brendan McKayn `nauty`[2]-aliohjelmakirjastoa, jolla voi laskea verkon kanonisen muodon tai automorfismiryhmän. `kotit3.c`:n alussa rivi

```
#include "nauty22/nauty.h
```

olettaa, että `kotit3.c` on hakemistossa, jossa on `nauty`-alihakemisto `nauty22`. Rivi

```
#define MAXN 60
```

olettaa, että nautyllä käsiteltävissä verkoissa on enintään 60 solmua. Aliohjelmassa `canon` multiverkosta tehdään verkko siten, että jokaisesta solmusta ja jokaisesta kaaresta tehdään solmu, joten tarkasteltavassa multiverkossa solmuja ja kaaria saa olla yhteensä korkeintaan `MAXN` – tämä riittäisi ainakin 20-hiilisten hiilivetyjen tarkasteluun, joten sen ei pitäisi olla rajoittava tekijä.

`kotit3.c` sisältää myös minimaalisen pääohjelman, joka kutsuu kerran `canon`-aliohjelmaa. Ainakin Linux-alustalla kääntäminen onnistuu komennolla:

```
cc -o kotit3 kotit3.c nauty22/nauty.c nauty22/nautil.c nauty22/naugraph.c
```

Palautus

Palautukset sähköpostitse assistentille (`Emilia.Oikarinen@hut.fi`) viimeistään maanantaina 26.4.2004 esimerkiksi hyvälaatuisena PostScriptinä. Vastauksessa tulee olla kunkin isomerialuokan koko, kuvaus käytetyistä menetelmistä, ohjelmien *kommentoitu* lähdekoodi sekä yhteenveto tarvitusta laskenta-ajasta esim. n:n mukaan eroteltuna.

[2] http://cs.anu.edu.au/~bdm/nauty/